

Programmer Manual

Table Class

1. Class Description

A table is an unordered group of “records”, each of which has a designated field called a key field. Records within the table are identified by the value of the key field. The types used for keys and values are specified using templates.

2. Class standings

Private Sections:

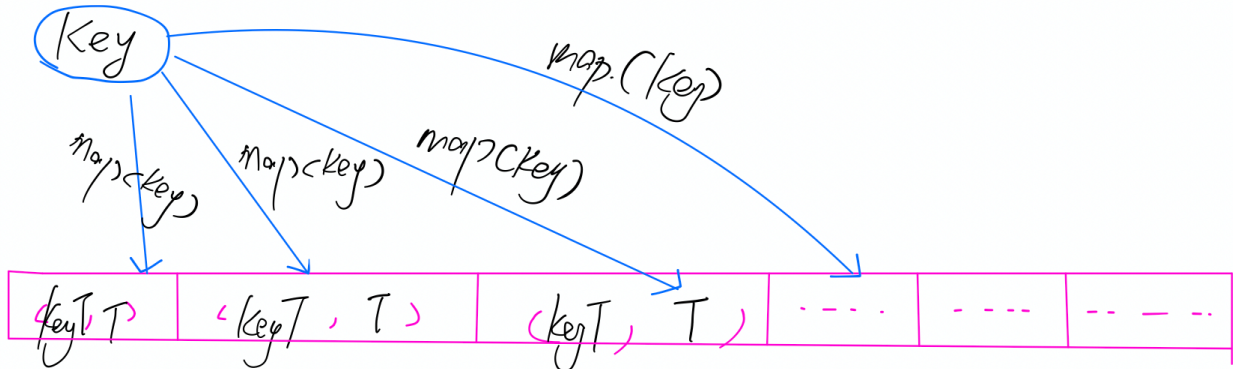
(int)tableCapacity: total size of the table
(int)tableSize: current allocated size of the table
(Pair*) the_table: a dynamic array pointer to table array of key-value Pairs.

Public Sections:

Table(int n, int (*map)(Key k))	/* Constructor */
~Table()	/* Destructor */
Table(const Table& initTable)	/* Copy Constructor */
operator=()	/* Operator */
insert()	/* Insert a key into table */
remove()	/* Remove a key from table */
lookUp()	/* Look up a key in table */
isIn()	/* Return true if table contains a key */
empty()	/* Return true if table is empty */
size()	/* Return current size of the table */
full()	/* Return true if the table is full */

3. High Level Program Solution

Internal Structure of Table Class



The table uses a one-dimension array structure internally. Because of this choice of internal representation, the class use a mapping function to map keys to an array index.

4. Limitations and Suggestions:

One of the biggest limitations is that the size of the table array cannot be resized as the table grows, if user needs to insert more elements into table when it's full, a new table with updated size is needed. This may cost time and can be slow.